Workflow Manager handling of the WDF

The following Workflow Description File contains all syntax elements defined so far :

Workflow Description File sample

```
workflow:
  name: My_Workflow

services:
  - name: ephemeral_service_1
    type: NFS
    attributes:
      namespace: "{{ NS1 }}"
      mountpoint: "/mnt/USER/My_Workflow/{{ SESSION }}/ES1"
      flavor: medium
  - name: ephemeral_service_2
    type: NFS
    attributes:
      namespace: "{{ NS2 }}"
      mountpoint: "/mnt/USER/My_Workflow/{{ SESSION }}/ES2"
      flavor: medium
    datamovers:
      - name: datamover1
        trigger: step_start
        target: flash
        operation: copy
        elements:
          - "gauges/*.hdf5"
          - "input/*"
      - name: datamover2
        trigger: step_stop
        target: tape
        operation: move
        elements:
          "gauges/*.hdf5"

steps:
  - name: step_A
    location:
      - gpu_module
    command: "sbatch My_Step_A"
    services:
      - name: ephemeral_service_1 RO
  - name: step_B
    location:
      - gpu_module
    command: "sbatch My_Step_B"
    services:
      - name: ephemeral_service_2 RW
        datamovers:
          - datamover1
          - datamover2
```

For each Workflow Manager command and each individual statement, we will list the actions to be taken by the workflow manager :

iosea_wf start

Action	CLI	Workflow Manager	Orchestrator
check the mandatory parameters : WORKFLOW= SESSION= Check that the WDF file can be read	X		
Identify the user	X	X	
Check if SESSION name is unique for this user		X	

check the syntax of the WDF • all sections are OK • variables are defined		X	
connect to keycloak and authenticate the user to get an S3 access token for each dataset	X		
acquire the credentials to submit jobs to Slurm on behalf of the user	X		
for each dataset-namespace required in the WDF • create the ephemeral service • allocate datanodes resources • record the service and connection information in the database			X

iosea_wf run

Action	CLI	Workflow Manager	Orchestrator
check the mandatory parameters : SESSION= STEP=	X		
check the validity of the parameters • the SESSION owned by the user is active • the STEP is described in the WDF		X	
for each required ephemeral service : • if a datamover with trigger step_start is specified ◦ launch the data transfer (implemented as a Slurm job ?) ◦ monitor the progress of the data transfer, update the status table of the session for the status command and wait until it is done • if an access mode hint is specified ◦ ignore for now 🙂			X
submit the step adding the ephemeral service connection option on the Slurm CLI			X
Monitor the progress of the job and update the status table for the "status" command until the job is done			X
for each required ephemeral service : • if a datamover with trigger step_stop is specified ◦ launch the data transfer (implemented as a Slurm job ?) ◦ monitor the progress of the data transfer, update the status table for the status command and wait until it is done			X

iosea_wf stop

Action	CLI	Workflow Manager	Orchestrator
check the mandatory parameters : SESSION=	X		
check the validity of the parameters • the SESSION owned by the user is active		X	
• check there is no active job (if yes, wait until they complete)			X
• check there is no active interactive access environment (if yes, wait until they complete)			X
• for each dataset-namespace required in the WDF ◦ delete the ephemeral service			X

iosea_wf access

Action	CLI	Workflow Manager	Orchestrator
check the mandatory parameters : SESSION=	X		
check the validity of the parameters • the SESSION owned by the user is active		X	
• launch a salloc SLURM command with options to attch to all ephemeral services			X
• update the status table of the session for the "status" command			X

Pending questions :

- add options to let users ask for cores/nodes resources
- allow sysadmins to define an interactive default partition

iosea_wf status

Action	CLI	Workflow Manager	Orchestrator
check the mandatory parameters : SESSION=	X		
check the validity of the parameters • the SESSION owned by the user is active		X	
• Display terminated steps • Slurm jobs terminated • data movers terminated			
• Display runing jobs • In Progress Data Movers • jobs submitted to Slurm, but not runing yet • jobs submitted to Slurm and running			
• Display active ephemeral services			

iosea_wf list

Action	CLI	Workflow Manager	Orchestrator
call Workflow Manager API	X		
report active sessions of the user		X	

Still to be defined :

Session sharing by different users

several possibilities

- token generated by the session owner and "given" to friends

- explicit autorisation activated by the session owner to other Workflow Manager users (as the WM will likely have to authenticate users)
- S3 credentials : any user requesting access to an active session owned by a different user having access to the S3 buckets (datasets) is allowed to "access" the session (iosea-wf access, iosea-run ?)
    - would be an additional option of the start command : iosea-wf start USER=xx SESSION=yy

Cancel command

to stop operations requested by a "run" command

- running or submitted Slurm jobs can be "scancelled", but what if the user wants to stop a huge data-mover operation in progress ?

Site Specific parameters

Many parameters are site specific, and are needed by the WP2 components :

- Module names (CPU, GPU, BigMem ...)
- data nodes list with their connections to Modules
- Ephemeral Services flavors (predefined sets of resources to be allocated to ephemeral services instances)
- Storage tiers (used in "target" statement
- tools to move data from one tier to another